

CECS 544
Semester Project
Summer 2024

I. Teams

Teams may consist of two students. A student may be on only one team. Switching teams is not allowed. All team members are responsible for every aspect of the project.

Please have your team member selection complete by July 12. First documentation is due Jul 17.

The project is sort of big. I don't recommend working alone, but if you want to do the project solo, that's okay.

"My teammate let me down" is not an acceptable excuse for poor performance on the project. If your teammate is not contributing to your satisfaction, let me know as early as possible.

II. Grading.

The total value of the project is 100 points. Each member of the team shares the total points earned. For example, if a team earns 100 points, each member gets 100 points in the calculation of their course grade.

Points are awarded on the following basis:

- Completeness – the project must be completed. All documentation must be turned in, and the software must function correctly and perform all the tasks indicated in the requirements. Missing documentation will incur a severe penalty on the team. Incomplete documentation shall incur a loss of points. Incomplete functionality in the code will incur a loss of points commensurate with the shortfall.
 - Correctness – the project must be correct. Documentation must be correct in what it states. The documentation must be accurate and sensible. The documentation must be free of ambiguity. The software must function correctly. All required tasks must be implemented. The software must be demonstrated as part of the team's presentation to me. Also, the documentation must be grammatically correct, with no spelling errors, and in Standard English.
 - Timeliness – All deliverables must be completed as the directed by the instructor on the dates required. Late deliverables will incur a severe penalty. In any event, NO deliverable shall be accepted after start of class.
-
- **An SQL or SQLite database is required for the application.**
 - **The program must have a GUI for all user accessible functions.**
 - **Browser based players are not permitted.**
 - **You may use whatever language(s) you like as long as you meet the requirements and use an external database.**

Our project – a cheap knock off of Apple itunes called MyTunes. The following are expected fetures of the product:

- Basic features of the GUI
 - *buttons for the following actions:*
 - *play a song from mp3 file, **
 - *stop playing a song, **
 - *pause playing a song, **
 - *un-pause playing a song, **
 - *skip to the next song (and play it) in the library, **
 - *skip to the previous song (and play it) in the library, **
 - *these are placed at the bottom of the GUI*
 - *a panel containing a list of all files in the song library and displays this information about each song:*
 - *song title*
 - *song artist*
 - *song album*
 - *song year*
 - *song genre*
 - *comment field which is editable*
 - *(all this data can be found in the mp3's IDTag (usually – it needs to have been stored there previously). You may need to research IDTags)*
 - *and the ability to:*
 - *add a song to the library, **
 - *delete a song from the library, **
 - *drag and drop (on the table) one or more songs that are added to the automatically database, **
- As well as these undocumented features:
 - *produce a popup menu that contains entries to add or delete a song, (by right-clicking anywhere in the Library area.*
 - *standard File menu bar entries for:*
 - *opening (and playing) a song not in the library,*
 - *exit the application,*
 - *add a song,*
 - *delete a song. (These features are the same as the popup menu above and do not require separate code. Simply link the menu button to the handler that is used for the popup menu items.*
 - *an sql or sqlite database that supplies mytunes with ALL data about songs stored via the GUI including tables for songs (and all associated IDTags), playlists and any other stored data for the program. ***
- **CODE THE ABOVE FEATURES FIRST. HAVE THESE DONE BY July 31. INITIAL TESTING OF THESE FEATURES WILL BE DONE July 31.**
- **Once the above features are completed, do these (by all means, you may add these features as you code the above but they will be tested later):**
 - *create a panel on the left or right side of the songs library in the GUI that is approximately 1/10 to 1/8 the width of the library table*

- in the new panel place two components
 - a tree with only one branch called "Library". Selecting (mouse) this branch shows the songs library in the main library table. This branch cannot expand or collapse and has no children.
 - a collapsible tree with root name "Playlist". Selecting this branch (double clicking) collapses and expands the tree alternatively. This tree expands to at most one level. That is, root can have children, but the children cannot. The children of the root node are playlists.
- allow users to create playlist by selecting "Create Playlist" from the File menu. Allow users to name it. Create it, then automatically select it in the tree and open it in the main library table component. The new playlist will be empty. All new playlist must originate in the main library window table. Playlist names and contents must be in the relational database and accessed via SQL. Playlist must show all six mp3v1 tags as should the library table.
- allow users to right click on a playlist. This action shows a popup context menu with "Open in New Window". Selecting this causes the playlist (empty or not) to be moved to a new window (complete with audio controls and menu but NO playlist or library trees) AND the main library table to be refilled with the song library (complete with audio controls and library and playlist trees)
- allow users to add songs from the library to an existing playlist in the following three ways:
 - select a song from the library and right click. A context menu (the same one from Iteration 1) has a new entry called "Add to playlist". Mousing over this entry causes a list of all playlists to appear in a sub-menu. Select the playlist to add the song.
 - Drag and drop from *outside* of myTunes on to the playlist in either of the following ways:
 - if the playlist is in the main library window, simply drag song files into the playlist
 - if the playlist is opened in a separate window, simply drag the songs into the playlist
 - in either event, if the song(s) being dragged onto a playlist is not in the library, add it to the library (without asking the user) before putting it in the playlist. In no event shall a song be in a playlist and not be in the library
- Drag and drop from *inside* of myTunes. This feature is only possible when a playlist is opened in a separate window and is initiated only from the library. Playlist to playlist transfers are not required, but are permitted if your team wishes to implement it. One or more files may be dragged.
- Right click on a playlist tree playlist show a popup context menu with "Delete Playlist". Selecting this deletes the highlighted playlist from the library (after confirmation), deletes its entry from the playlist-song relational table and redraw the playlist tree.

- Add a Volume slider. Slider should affect only its window and not other open windows.
- ***After completing the above, add these features:***
 - Take the "File" column out of the library display (it may remain in the table, just don't display it)
 - Make the "Title" column permanent - that is, always displayed.
 - Columns "Album", "Artist", "Year", "Genre" and "Comment" are to be made selectable (displayed or not). This is accomplished through a context sensitive popup menu accessed by right clicking the table headers. Each of the selectable fields are listed (permanently) in the popup and each has a "checkbox" next to the column name. A "checked" checkbox means the column is displayed, and an "unchecked" checkbox means the column is not displayed.
 - Clicking on the column name in the popup toggles the column to display or not.
 - The column configuration must persist between Playlists and Library (that is, have the same configuration) and must persist from session to session - the application must display the same configuration on starting as it had on the previous session when closed.
 - Default sorting is on "Title". For a 5 point bonus, sort on any column (and maintain correct functionality).
 - Two timers and a progress bar must be displayed. Location of these elements is at your discretion, but the three elements must be side by side.
 - The "left" timer displays elapsed time since the song began playing, in H:MM:SS
 - The "right" timer displays the song's length minus the elapsed time, in H:MM:SS
 - The progress bar (length, style, color also at your discretion) moves along with the passing of time as a song plays. At the commencing of play, the bar is "empty". The "ticks" need not be uniform but something approaching uniform. When a song is completed, the bar should be filled and then returned to empty.
 - A "Controls" menu is added. It contains the following commands:
 - "Play" - selecting this plays the currently selected song. If no song is selected, play the first in the library. If "Shuffle" is on, play a random selection. Accelerator key is "Space".
 - "Next" - plays the next song in the library. Ties to the "Skip Forward" function. Accelerator key is "Ctrl-RightArrow".
 - "Previous" - plays the previous song in the library. Ties to the "Skip Backwards" function. Accelerator key is "Ctrl-LeftArrow".
 - "Play Recent" - a submenu appears upon **mouse over** of "Play Recent" and displays up to the last ten songs played by overt selection (click song/play, click "Skip" forward or backward). Selecting a song from the list plays the song and puts it in "Recent Play" list again. Songs played by "Shuffle" are not put in the recent play list. If less than ten songs have been played, display as

many as have been played. Recent Play list must persist the session. (The same songs must be in the list at startup as was in the list at shutdown.)

- "Go to Current Song" - highlight the currently playing song and scroll the screen to insure that song is visible. If no song is playing, but one is selected, scroll the screen to insure it is visible. If no song is selected, do nothing. Accelerator key is "Ctrl-L".
- A separator.
- "Increase Volume" - ties to the volume slider. 5% increase for each click. Accelerator key is "Ctrl-I". (Slider must move as volume increases.)
- "Decrease Volume" - ties to the volume slider. 5% decrease for each click. Accelerator key is "Ctrl-D". (Slider must move as volume decreases.)
- A separator.
- "Shuffle" - with a checkbox. Checkbox not selected, command has no affect. Checked, the command picks a song at random from the library and plays. If a song is currently playing, command has no effect until current song completes. Shuffle continues to randomly select and play until unchecked. Unchecking has not effect on currently playing song. No Accelerator.
- "Repeat" - repeat the currently playing song forever. If no song is playing, repeat forever the next song that is played. Turning off "Repeat" ends the effect after the currently playing song ends. No accelerator.

- **DELIVERABLES**

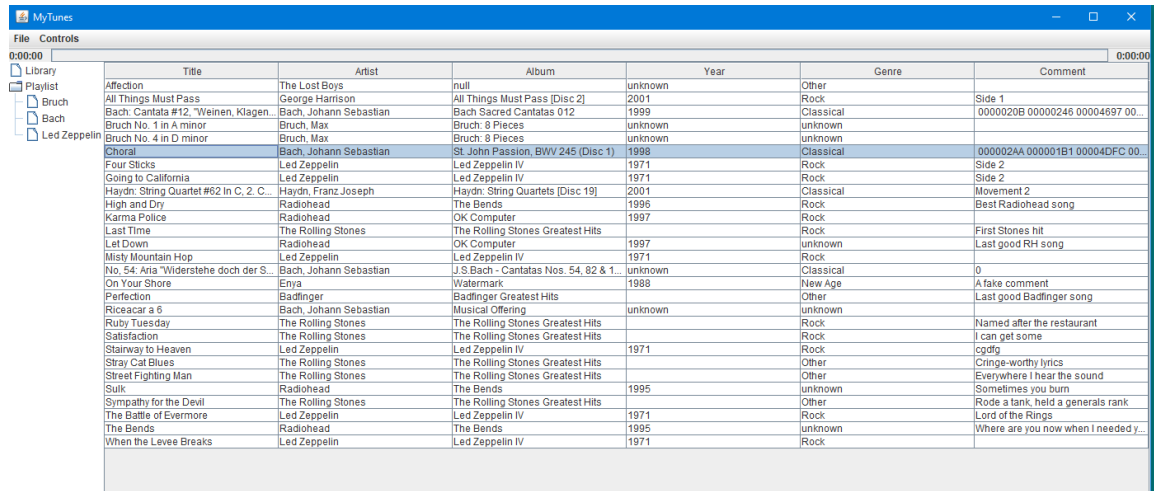
- Requirements documentation due July 17:
 - initial set of user stories (XP modeling) on all features marked with *.
 - UML component diagram
 - initial UML class diagram
- July 31 – working program for features up to **
- Aug 12 – all remaining features plus updated UML class diagram.

Final testing will commence Aug 12.

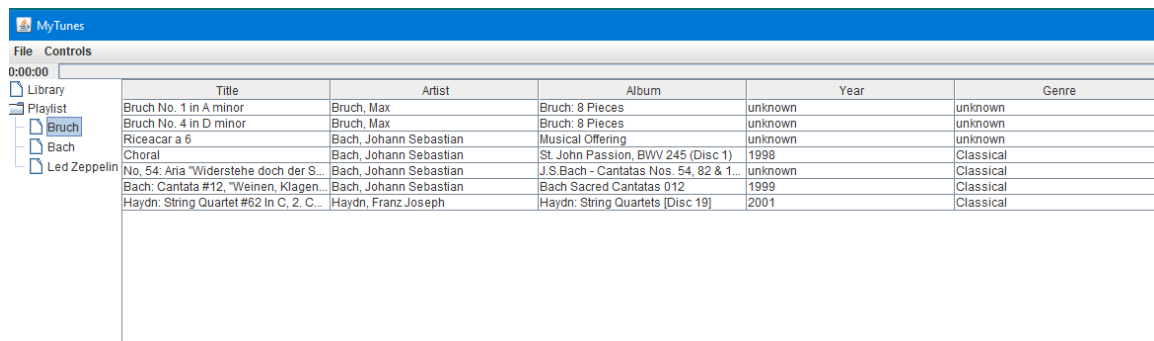
Procedures.

All documentation must be typed (word processed). All figures, drawing, charts, etc. must be original, and include no hand drawing. The **team member's names** must be on the title page. **Documents must have a table of contents, pages numbers and a list of figures. All charts, figures, etc must have a border and a caption.** User story cards may be hand written but must be scanned and placed in your document as an image.

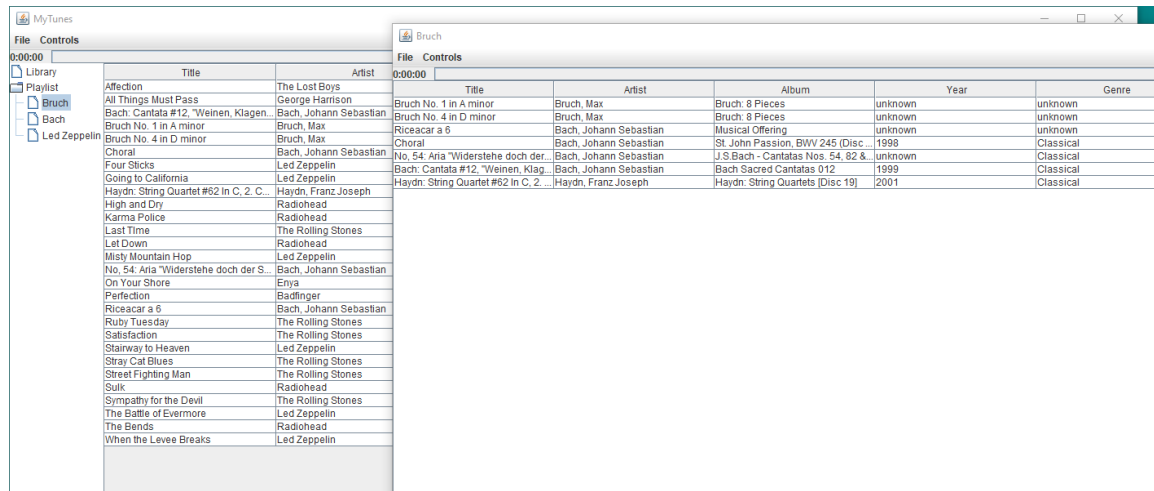
Appendix: Images



General appearance of the GUI



GUI with playlist selected



GUI with playlist opened in separate window

The screenshot shows the MyTunes application interface. On the left, a sidebar contains 'Library' and 'Playlist' sections. The 'Playlist' section is expanded, showing a tree view with 'Bruch', 'Bach', and 'Led Zeppelin'. The main area displays a table of songs with columns for Title, Artist, Album, and a status column. A context menu is open over the song 'High and Dry' by Radiohead, showing options: 'Add song to Library', 'Add to Playlist', and 'Delete currently selected song'. The 'Add to Playlist' option is further expanded, showing sub-options for 'Bruch', 'Bach', and 'Led Zeppelin'.

Title	Artist	Album	Status
Affection	The Lost Boys	null	unkr
All Things Must Pass	George Harrison	All Things Must Pass [Disc 2]	200
Bach: Cantata #12, "Weinen, Klagen...	Bach, Johann Sebastian	Bach Sacred Cantatas 012	1999
Bruch No. 1 in A minor	Bruch, Max	Bruch: 8 Pieces	unkr
Bruch No. 4 in D minor	Bruch, Max	Bruch: 8 Pieces	unkr
Choral	Bach, Johann Sebastian	St. John Passion, BWV 245 (Disc 1)	1999
Four Sticks	Led Zeppelin	Led Zeppelin IV	197
Going to California	Led Zeppelin	Led Zeppelin IV	197
Haydn: String Quartet #62 In C, 2. C...	Haydn, Franz Joseph	Haydn: String Quartets [Disc 19]	200
High and Dry	Radiohead	The Bends	1999
Karma Police	Radiohead	OK Computer	1999
Last Time	The Rolling Stones	The Rolling Stones Greatest Hits	
Let Down	Radiohead	OK Computer	1999
Misty Mountain Hop	Led Zeppelin	Led Zeppelin IV	197
No. 54: Aria "Widerstehe doch der S...	Bach, Johann Sebastian	J.S.Bach - Cantatas Nos. 54, 82 & 1...	unkr
On Your Shore	Enya	Watermark	1988
Perfection	Badfinger	Badfinger Greatest Hits	
Riceacar a 6	Bach, Johann Sebastian	Musical Offering	unkr
Ruby Tuesday	The Rolling Stones	The Rolling Stones Greatest Hits	
Satisfaction	The Rolling Stones	The Rolling Stones Greatest Hits	
Stairway to Heaven	Led Zeppelin	Led Zeppelin IV	197
Stray Cat Blues	The Rolling Stones	The Rolling Stones Greatest Hits	
Street Fighting Man	The Rolling Stones	The Rolling Stones Greatest Hits	
Sulk	Radiohead	The Bends	1999
Symnathy for the Devil	The Rolling Stones	The Rolling Stones Greatest Hits	

MyTunes with adding to playlists via popup menus.

The screenshot shows the MyTunes application interface. On the left, a sidebar contains 'Library' and 'Playlist' sections. The 'Playlist' section is expanded, showing a tree view with 'Bruch', 'Bach', and 'Led Zeppelin'. The main area displays a table of songs with columns for Title, Artist, and Album. A context menu is open over the song 'All Things Must Pass' by George Harrison, showing options: 'Open in new window' and 'Delete Playlist'.

Title	Artist	Album
Affection	The Lost Boys	null
All Things Must Pass	George Harrison	All Things Must Pass [Disc 2]
Bach: Cantata #12, "Weinen, Klagen...	Bach, Johann Sebastian	Bach Sacred Cantatas 012
Bruch No. 1 in A minor	Bruch, Max	Bruch: 8 Pieces
Bruch No. 4 in D minor	Bruch, Max	Bruch: 8 Pieces
Choral	Bach, Johann Sebastian	St. John Passion, BWV 245 (Disc 1)
Four Sticks	Led Zeppelin	Led Zeppelin IV
Going to California	Led Zeppelin	Led Zeppelin IV
Haydn: String Quartet #62 In C, 2. C...	Haydn, Franz Joseph	Haydn: String Quartets [Disc 19]
High and Dry	Radiohead	The Bends
Karma Police	Radiohead	OK Computer
Last Time	The Rolling Stones	The Rolling Stones Greatest Hits
Let Down	Radiohead	OK Computer
Misty Mountain Hop	Led Zeppelin	Led Zeppelin IV
No. 54: Aria "Widerstehe doch der S...	Bach, Johann Sebastian	J.S.Bach - Cantatas Nos. 54, 82 & 1...
On Your Shore	Enya	Watermark
Perfection	Badfinger	Badfinger Greatest Hits
Riceacar a 6	Bach, Johann Sebastian	Musical Offering
Ruby Tuesday	The Rolling Stones	The Rolling Stones Greatest Hits
Satisfaction	The Rolling Stones	The Rolling Stones Greatest Hits
Stairway to Heaven	Led Zeppelin	Led Zeppelin IV